DWF_dp_rnd functions

Arithmetic Rounding

Version, STAR, and myDesignWare Subscriptions: IP Directory

Description

The DWF_dp_rnd functions truncate the lower bits of argument `a` below the bit position specified by argument `lsb` and return a rounded value according to the rounding mode specified by argument `mode`. Argument `a` and the return value are both either signed (two's complement) or unsigned.

Table 1-1 Function Names

Function Name	Description
DWF_dp_rnd	VHDL unsigned rounding
DWF_dp_rnd	VHDL signed (two's complement) rounding
DWF_dp_rnd_uns	Verilog unsigned rounding
DWF_dp_rnd_tc	Verilog signed (two's complement) rounding

Table 1-2 Argument Description

Argument Name	Type	Width / Values	Description
a	Vector	width	Input data
lsb	Integer	≥ 1, ≤ width - 1	LSB index of return value (VHDL only, constant)
mode	Vector	4	Rounding mode (values in Table 1-4 on page 2)
DWF_dp_rnd	Vector	width - lsb	Return value

Table 1-3 Parameter Description (Verilog)

Parameter	Values	Description
width	≥ 1	Word length of input `a`
lsb	≥ 1, ≤ width - 1	LSB index of return value

Verilog Include File: `DW_dp_rnd_function.inc`

Functional Description

Input `a[width-1:0]` is divided into an integer part `a[width-1:lsb]` and a fractional part `a[lsb-1:0]` with the property

```
a[width-1:lsb] ≤ a[width-1:lsb].a[lsb-1:0] < a[width-1:lsb]+1
```

Rounding either returns the integer part (`a[width-1:lsb]`) or its increment (`a[width-1:lsb]+1`) depending on the fractional part (`a[lsb-1:0]`) and a threshold defined by the rounding mode.

```
z[width-lsb-1:0] = DWF_dp_rnd (a[width-1:0], lsb, mode)
```

Effectively, the following occurs:

```
if a[lsb-1:0] > threshold(mode)
        z[width-lsb-1:0] = a[width-1:lsb]+1
else
        z[width-lsb-1:0] = a[width-1:lsb]
```

Rounding Modes

The DWF_dp_rnd functions provide a comprehensive set of rounding modes that are summarized in Table 1-4. The rounding mode is selected by assigning argument `mode` either a predefined constant value or a 4-bit binary vector (dynamic).

The rounding modes differ with respect to rounding direction and tie resolution (a tie is the value `a[width + 1:lsb].1` that lies exactly in the middle between `a[width - 1:lsb]` and `a[width - 1:lsb] + 1`). Rounding modes can be classified in terms of rounding symmetry, error bounds, bias (= accumulated error over many random samples) and hardware complexity, as shown in Table 1-5 on page 3.

Table 1-4 Rounding Modes

Predefined Constant [a]	Value [b]	Rounding Direction	Tie Resolution	Class
DW_dp_rnd_neg_inf [c]	0000	Towards neg. infinity	None	A
DW_dp_rnd_pos_inf [c]	0001	Towards pos. infinity	None	B
DW_dp_rnd_zero [c]	0010	Towards zero	None	C (signed) A (unsigned)
DW_dp_rnd_away	0011	Away from zero	None	C (signed) B (unsigned)
DW_dp_rnd_even	0100	Towards even	None	C
DW_dp_rnd_odd	0101	Towards odd	None	C
DW_dp_rnd_near_neg_inf	1000	Towards nearest	Towards neg. infinity	D
DW_dp_rnd_near_pos_inf	1001	Towards nearest	Towards pos. infinity	E
DW_dp_rnd_near_zero	1010	Towards nearest	Towards zero	F (signed) D (unsigned)

Table 1-4 Rounding Modes (Continued)

Predefined Constant [a]	Value [b]	Rounding Direction	Tie Resolution	Class
DW_dp_rnd_near_away	1011	Towards nearest	Away from zero	F (signed) E (unsigned)
DW_dp_rnd_near_even [c]	1100	Towards nearest	Towards even	F
DW_dp_rnd_near_odd	1101	Towards nearest	Towards odd	F

[a] Predefined constant value to be assigned to argument `mode`.

[b] Binary vector to be assigned to argument `mode`. The individual bits can be interpreted as follows:
- `mode[3]` : nearest : 0 = not towards nearest, 1 = towards nearest
- `mode[2:1]` : symmetric : 00 = not symmetric, 01 = symmetric wrt zero, 10= symmetric wrt even
- `mode[0]` : direction : 0 = towards negative/0/even, 1 = towards positive/infinity/odd

[c] IEEE rounding modes (correspond to `floor`, `ceil`, `trunc` and `round` in the C programming language).

Table 1-5 Rounding Classes

Class	Symmetric	Error [a]	Bias	Complexity
A	no	$0 \leq \text{error} \leq 1 - 2^{-lsb}$	$(1 - 2^{-lsb})/2$	lowest
B	no	$-(1 - 2^{-lsb}) \leq \text{error} \leq 0$	$-(1 - 2^{-lsb})/2$	low
C	yes	$-(1 - 2^{-lsb}) \leq \text{error} \leq 1 - 2^{-lsb}$	0	highest
D	no	$-(1/2 - 2^{-lsb}) \leq \text{error} \leq 1/2$	$2^{-lsb}/2$	low
E	no	$-1/2 \leq \text{error} \leq 1/2 - 2^{-lsb}$	$-2^{-lsb}/2$	low
F	yes	$-1/2 \leq \text{error} \leq 1/2$	0	high

[a] The error given is relative to the rounded output.

For more information about the DesignWare datapath functions, refer to the topic titled Arithmetic – Datapath Functions Overview.

Related Topics

- Arithmetic – Datapath Functions Overview
- DesignWare Building Block IP User Guide

VHDL Example

```
library IEEE, DWARE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
use DWARE.DW_dp_functions.all;
-- DWARE.DW_dp_functions_arith package if IEEE.std_logic_arith is used

entity DWF_dp_rnd_test is
  port (a, b, c : in  signed(7 downto 0);
        z       : out signed(7 downto 0));
end DWF_dp_rnd_test;

architecture rtl of DWF_dp_rnd_test is
begin
  z <= DWF_dp_rnd (a * b, 8, DW_dp_rnd_near_even) + c;
end rtl;
```

Verilog Example

```
module DWF_dp_rnd_test (a, b, c, z);

  input  signed [7:0] a, b, c;
  output signed [7:0] z;

  // Passes the parameters to the function
  parameter width = 16;
  parameter lsb   = 8;

  // add "$SYNOPSYS/dw/sim_ver" to the search path for simulation
  `include "DW_dp_rnd_function.inc"

  assign z = DWF_dp_rnd_tc (a * b, DW_dp_rnd_near_even) + c;

endmodule
```

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com